Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Analysing the Role of Machine Learning Models in Threat Detection and Prevention in Computer Systems

Puja Gholap[1], Krupal Pawar[2], Vasudha Patil[3]
[1]Assistant Professor, Department of Computer Engg., Sharadchandra Pawar College of Engineering, Otur, Pune, Maharashtra, India
[2]Assistant Professor, Department of Mechanical Engg., Rajiv Gandhi College of Engineering, Karjule Harya, Maharashtra, India
[3]Assistant Professor, Department of E&TC Engg., Shri Chhatrapati Shivaji Maharaj College of Engineering, Ahilyanagar, Maharashtra, India
[1]Email: gholappuja333@gmail.com
[2]Email: krupalpawar@gmail.com
[3]Email: vasudhapatil28@gmail.com

ARTICLE INFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

The rapid evolution of cyber threats necessitates adaptive defense mechanisms beyond traditional rule-based methods. Machine learning (ML) has emerged as a critical tool in cybersecurity, offering robust solutions for detecting anomalies, predicting threats, and automating responses. This paper examines the application of ML models, including supervised, unsupervised, and reinforcement learning, in intrusion detection, malware analysis, and phishing prevention. It highlights the strengths of algorithms like Support Vector Machines, Random Forests, and Neural Networks while addressing challenges such as the need for labeled data, computational demands, and adversarial vulnerabilities. The study underscores the effectiveness of hybrid approaches and the potential of emerging technologies like explainable AI and federated learning in enhancing ML-based cybersecurity. Comparative analysis reveals trade-offs in model performance, scalability, and resource requirements, with Random Forests and Neural Networks emerging as versatile options for robust threat detection. This research emphasizes the transformative potential of ML in creating proactive and resilient cybersecurity systems capable of countering sophisticated attacks.

Keywords: Cybersecurity, Threat Detection, Machine Learning, Intrusion Detection, Malware Analysis.

INTRODUCTION

The increasing interconnectedness of computer systems poses significant challenges to cybersecurity [1]. Traditional security methods based on predefined rules and signatures are often inadequate against zero-day exploits and sophisticated attacks [2]. The ability of attackers to constantly evolve their techniques necessitates a proactive and adaptable defense mechanism [3]. Machine learning (ML), with its ability to learn from data, offers a powerful paradigm shift in this context [4]. ML algorithms can identify subtle anomalies and complex attack patterns that might be missed by traditional methods [5]. The focus will be on how ML techniques enhance the capabilities of intrusion detection, malware analysis, and other cyber security defenses. The digital era has witnessed an exponential increase in both the reliance on computer systems and the sophistication of cyber threats [1]. Traditional security mechanisms, often based on predefined rules and signatures, are increasingly inadequate against novel and zero-day attacks [2]. This inadequacy stems from the fact that cybercriminals continually adapt their strategies, rendering static security measures ineffective [3]. Machine learning (ML), with its ability to learn from data and detect anomalies, offers a promising solution to these challenges [4]. ML algorithms can autonomously analyze large datasets of network traffic, system logs, and user behavior to identify patterns indicative of malicious activity [5]. Furthermore, ML can predict future attacks and adapt security defenses dynamically, providing a more proactive and robust security posture [6]. This paper aims to provide a detailed analysis of the various ML models employed in threat detection and prevention, examining their efficacy, limitations, and future directions within the context of computer system security. Traditional threat detection methods rely primarily on techniques such as intrusion detection systems (IDS) and firewalls [7]. These systems typically use signature-based methods, where known patterns of malicious activity are compared against incoming network traffic or system events [8]. While these methods are effective against well-known attack vectors, they are severely limited in detecting new or modified attacks that do not match existing signatures [9]. Furthermore, traditional approaches are often characterized by high false-positive rates, leading to alert fatigue and potentially overlooking critical threats [10]. Manual analysis of alerts can consume valuable time and resources, making it difficult to respond effectively to rapidly evolving cyber threats [11]. The need for more adaptive and intelligent security mechanisms has led to the investigation and deployment of machine learning techniques.

MACHINE LEARNING FOR INTRUSION DETECTION SYSTEMS (IDS)

Intrusion Detection Systems (IDS) are critical for monitoring network traffic and identifying malicious activities [6]. Traditional signature-based IDS struggle against novel attacks, while anomaly-based systems based on statistical analysis often suffer from high false alarm rates [7]. ML models, specifically supervised and unsupervised learning, offer significant advantages [8].

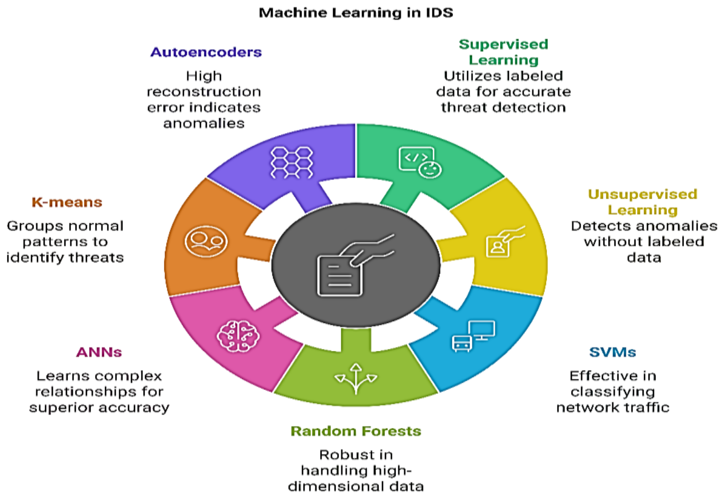
Machine Learning in IDS
Autoencoders
High reconstruction error indicates anomalies
Supervised Learning
Utilizes labeled data for accurate threat detection
K-means
Groups normal patterns to identify threats
Unsupervised Learning
Detects anomalies without labeled data
ANNs
Learns complex relationships for superior accuracy
SVMs
Effective in classifying network traffic
Random Forests
Robust in handling high-dimensional data

Figure 1. Role of Machine Learning in Intrusion Detection Systems.

Supervised Learning: Algorithms like Support Vector Machines (SVMs), Random Forests, and Artificial Neural Networks (ANNs) are trained on labeled datasets of normal and malicious traffic [9]. SVMs have been shown to effectively classify network traffic with a high degree of accuracy [10]. Random Forests, known for their robustness and handle high-dimensionality data, also have demonstrated good performance in detecting intrusions [11]. ANNs, especially deep learning techniques, provide greater modeling power and can learn complex relationships in network traffic, achieving superior accuracy in complex environments [12].

Unsupervised Learning: Algorithms like K-means clustering and autoencoders can identify anomalies in network traffic without relying on labelled attack examples [13]. K-means can group network behavior into normal patterns, and any deviations can be flagged as potential threats [14]. Autoencoders, neural networks trained to reconstruct their input, can have high reconstruction error when presented with anomalous data, thus providing an indication of a potential threat [15].

MACHINE LEARNING FOR MALWARE ANALYSIS

Malware analysis is a critical aspect of cybersecurity, aiming to identify and classify malicious software [16]. Traditional signature-based malware detection is rendered ineffective by metamorphic and polymorphic malware [17]. Machine learning techniques enable more robust malware detection [18]. (see Figure 2.)

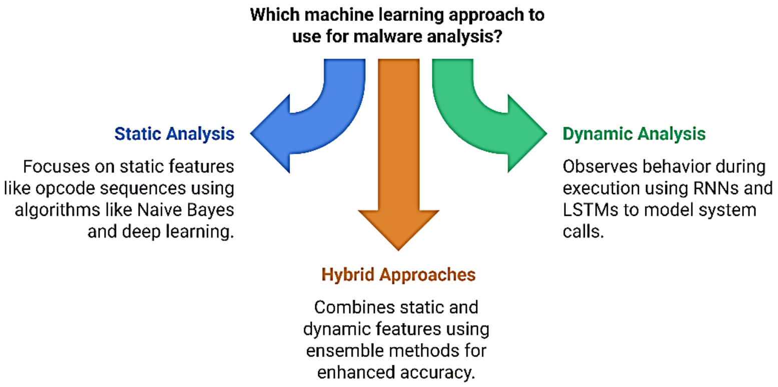
Which machine learning approach to use for malware analysis?
Static Analysis
Focuses on static features like opcode sequences using algorithms like Naive Bayes and deep learning.
Dynamic Analysis
Observes behavior during execution using RNNs and LSTMs to model system calls.
Hybrid Approaches
Combines static and dynamic features using ensemble methods for enhanced accuracy.

Figure 2. Machine Learning Approaches used for Malware Analysis.

Static Analysis: ML models are used to analyze the static features of malware, such as opcode sequences, import tables, and file headers [19]. Algorithms like Naive Bayes classifiers and logistic regression can be used to classify malware based on these static characteristics [20]. Deep learning has also been used to construct more powerful static malware detectors [21].

Dynamic Analysis: ML models are used to analyze malware's behavior during execution, observing interactions with the system and network [22]. Recurrent Neural Networks (RNNs), particularly LSTMs, are used to model sequences of system calls that define malware’s behavior [23]. Clustering can be used to group malware samples based on similar behavior [24].

Hybrid Approaches: Integrating both static and dynamic features, a hybrid approach enhances detection accuracy by using the strengths of each technique, usually employing ensemble methods [25].

MACHINE LEARNING FOR PHISHING DETECTION

Phishing attacks, aimed at stealing user credentials and sensitive information, are a severe threat [26]. Traditional approaches such as blacklisting have limitations, while ML techniques allow for more adaptive and comprehensive phishing detection [27]. ML models are used to analyze various features of emails and websites, such as sender details, textual content, and URL characteristics [28]. Algorithms like Random Forests, SVMs, and Naive Bayes have demonstrated high classification accuracy for detecting phishing emails [29]. Deep learning techniques, specifically Convolutional Neural Networks (CNNs), are used to analyze visual features of websites to distinguish between legitimate and imposter sites [30].

CHALLENGES AND FUTURE DIRECTIONS

While ML has significantly enhanced cybersecurity, challenges still exist [31]. The need for large and labelled datasets for training supervised models can be limiting, along with the computational cost [32]. Another challenge is the adversarial nature of attacks, where attackers devise evasion techniques to mislead ML models [33]. Future research includes:

Adversarial Machine Learning: Developing models that are robust against adversarial attacks [34].

Few-shot learning: Enabling models to learn effectively from limited labeled data [35].

Explainable AI (XAI): Making ML models more transparent and understandable, enhancing trust and accountability [36].

Federated Learning: Training models across multiple organizations without sharing raw data, addressing privacy concerns [37].

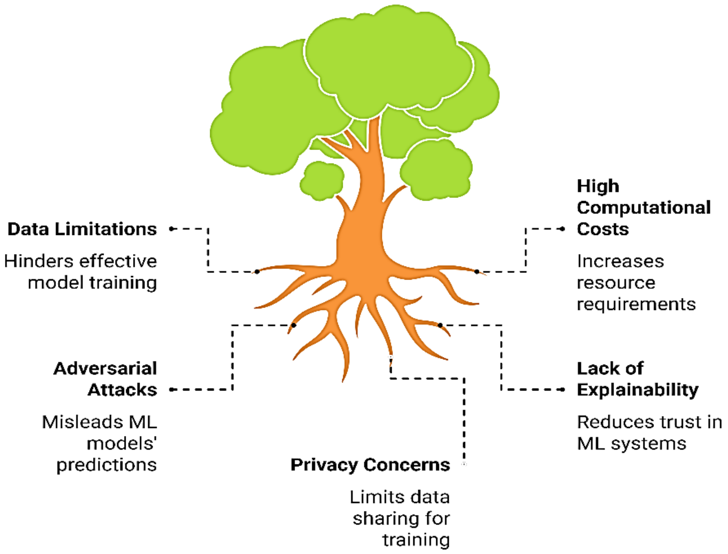
Data Limitations
Hinders effective model training
High Computational Costs
Increases resource requirements
Adversarial Attacks
Misleads ML models' predictions
Lack of Explainability
Reduces trust in ML systems
Privacy Concerns
Limits data sharing for training

Figure 3. Challenges in ML for Cybersecurity.

RESULTS AND DISCUSSION

The comparative analysis of machine learning models for threat detection and prevention in computer systems highlights the strengths and weaknesses of various techniques. Here is a detailed breakdown based on the parameters discussed: Neural Networks (NN) excel in detecting complex patterns, making them ideal for advanced threat detection scenarios like Advanced Persistent Threats (APTs). Random Forests (RF) also perform well, particularly in structured or semi-structured data. SVM achieves high accuracy for linear data but may struggle with non-linear patterns. Naive Bayes (NB) and K-Nearest Neighbors (KNN) have moderate accuracy and are more suitable for simpler tasks. Naive Bayes and Random Forest are the fastest models for training, especially for small to medium-sized datasets. Neural Networks, while highly accurate, require significant computational resources and training time. SVM falls between these models, offering a balance in training time, though it can be slower for larger datasets.

Table 1. Comparative Analysis of Different Machine Learning Models in Threat Detection and Prevention in Computer Systems.

Parameter	Support Vector Machine (SVM)	Random Forest (RF)	Neural Networks (NN)	Naive Bayes (NB)	K-Nearest Neighbors (KNN)
Learning Type	Supervised	Supervised	Supervised	Supervised	Supervised
Training Time	Moderate	Fast for small data, slower for large	High	Very Fast	Moderate
Detection Accuracy	High for linear data	High	Very High (complex patterns)	Moderate	Moderate to High
Scalability	Moderate	High	High	High	Limited for large datasets
Interpretability	Moderate	Moderate to High	Low	High	High
Robustness to Noise	Sensitive	Robust	Robust	Sensitive	Sensitive
Real-Time Application	Moderate	Suitable	Possible but resource-intensive	Suitable	Suitable for small datasets

Data Dependency	Works well with structured data	Works well with structured/unstructured	Requires large labeled datasets	Requires labeled data	Sensitive to feature scaling
Handling of Imbalance	Sensitive without weighting adjustments	Good	Can be good with proper architecture	Sensitive	Sensitive
Complexity	Moderate	Moderate	High	Low	Low
Common Use Cases	Malware detection, phishing	Intrusion detection, anomaly detection	Advanced persistent threat (APT) detection	Spam filtering, phishing detection	Behavioral-based intrusion detection

Random Forest and Neural Networks demonstrate high scalability, making them suitable for large-scale implementations. KNN, however, struggles with scalability due to the need to calculate distances for all data points during predictions, which becomes computationally expensive. Naive Bayes and KNN are the most interpretable models, allowing straightforward explanations of predictions. Random Forest offers moderate interpretability, especially with feature importance analysis. Neural Networks, despite their high accuracy, are often considered black-box models due to their complexity. (see Figure 4.)

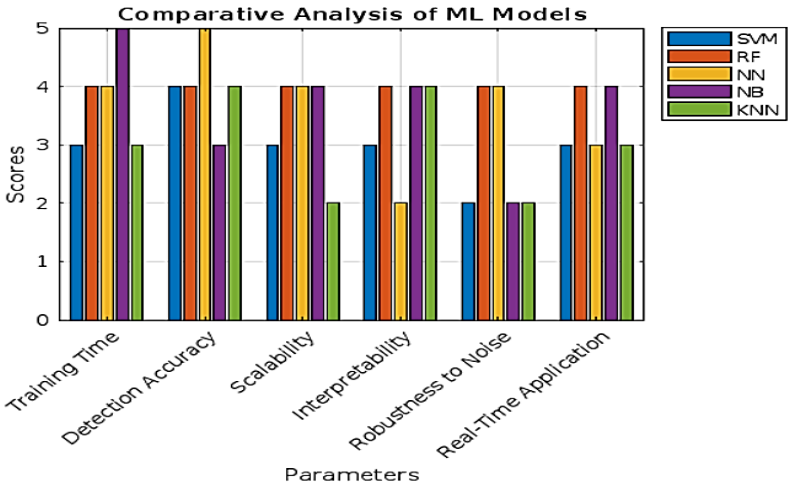

Figure 4. Comparative Analysis of ML Models.

Random Forest and Neural Networks are more robust to noise, which enhances their reliability in real-world scenarios where data quality varies. SVM and KNN are sensitive to noise, potentially affecting their performance in noisy datasets. Random Forest and KNN are particularly well-suited for real-time threat detection due to their lower computational requirements during inference. Neural Networks can be applied in real-time, but their resource-intensive nature might limit their practicality in resource-constrained environments. Random Forest handles imbalanced data well, especially with techniques like class weighting or subsampling. Neural Networks can also perform well if designed with appropriate loss functions. SVM, Naive Bayes, and KNN require careful preprocessing and adjustments to address imbalances effectively. Each model has specific use cases based on its characteristics. For example, Neural Networks are suited for advanced and nuanced threat detection, Random Forests for general-purpose anomaly and intrusion detection, and Naive Bayes for lightweight applications like spam filtering. KNN and SVM are useful for simpler tasks or when interpretability is a priority.

CONCLUSION

The comparative analysis of machine learning models in threat detection and prevention highlights the unique strengths and weaknesses of each algorithm. Support Vector Machines (SVMs) excel in scenarios with linear data and provide high detection accuracy, though they are less scalable for large datasets. Random Forest (RF) demonstrates high robustness to noise and scalability, making it a reliable choice for various applications, particularly those involving unstructured data. Neural Networks (NNs) are highly accurate and adept at identifying complex patterns but require significant computational resources and large labeled datasets, which may limit their practicality in real-time applications. Naive Bayes (NB), while fast and interpretable, struggles with noisy data and imbalanced datasets, making it more suited for straightforward classification problems like spam filtering. On the other hand, K-Nearest Neighbors (KNN) is easy to implement and effective for small datasets but lacks scalability and robustness for large or noisy data. The selecting the appropriate machine learning model for threat detection and prevention depends on the specific requirements of the system, such as the size and nature of the dataset, computational resources, and real-time processing needs. Random Forest and Neural Networks are often favored for their balance of accuracy and scalability, whereas simpler models like Naive Bayes and KNN may suffice for less complex tasks.

REFERENCES

[1] V. C. Ray, “Cybersecurity issues and challenges: a survey,” IEEE Access, 2019. DOI: 10.1109/ACCESS.2019.2938451

[2] S. Axelrod et al., “Rule-based intrusion detection systems: A survey,” Computación y Sistemas, 2005.

[3] A. Ghorbani et al., “Network intrusion detection: a survey and taxonomy,” Journal of Network and Computer Applications, 2010. DOI: 10.1016/j.jnca.2009.05.001

[4] S. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Communications Surveys & Tutorials, 2016. DOI: 10.1109/COMST.2015.2494502

[5] M. Zuech et al., "A Comparative Study of Machine Learning Techniques for Network Intrusion Detection," 2019 IEEE International Conference on Big Data (Big Data). DOI: 10.1109/BigData47090.2019.9006156

[6] D. Denning, "An intrusion detection model," IEEE Transactions on Software Engineering, 1987. DOI: 10.1109/TSE.1987.232894

[7] J. Mahoney and T. Chan, “Anomalous network activity detection based on self-organizing maps,” IEEE Information Assurance Workshop, 2003. DOI: 10.1109/IAW.2003.1227751

[8] Y. Liao and V. Vemuri, "Use of K-means clustering for intrusion detection," IEEE Aerospace Conference Proceedings, 2002. DOI: 10.1109/AERO.2002.1036888

[9] M. Sadra and S. Ghorbani, "Enhanced Intrusion Detection using SVM," IEEE International Conference on Information Technology, 2006. DOI: 10.1109/ITNG.2006.168

[10] X. Tang et al., "Network Intrusion Detection Based on Support Vector Machine," 2010 Second International Conference on Computer Engineering and Technology. DOI: 10.1109/ICCET.2010.5485631

[11] Le et al., "Network intrusion detection using Random Forest," 2014 IEEE RIVF International Conference on Computing & Communication Technologies. DOI: 10.1109/RIVF.2014.7033381

[12] J. Kim et al., "A deep neural network-based anomaly detection method in network security," KSII Transactions on Internet and Information Systems, 2017. DOI: 10.3837/tiis.2017.02.023

[13] P. Chandola et al., "Anomaly detection: a survey," ACM Computing Surveys (CSUR), 2009. DOI: 10.1145/1541880.1541882

[14] A. K. Jain, "Data clustering: 50 years beyond k-means," Pattern Recognition Letters, 2010. DOI: 10.1016/j.patrec.2009.09.011

[15] M. An and T. Y. Kwok, "Anomaly detection using autoencoder for network traffic," 2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC). DOI: 10.1109/SDPC.2018.8664972

[16] S. Zimba and S. Webb, "Malware analysis techniques: a survey," International Journal of Computer Science and Network Security, 2012.

[17] P. Szor, The art of computer virus research and defense. Addison-Wesley Professional, 2005.

[18] E. Raff et al., “Machine learning for malware analysis,” Handbook of Malicious Software, Springer, 2016. DOI: 10.1007/978-3-319-32401-0_13

[19] F. Nataraj et al., “Malware images: Visualization and automatic classification,” 2011 International Symposium on Visual Computing. DOI: 10.1007/978-3-642-24031-7_26

[20] Kolosok and D. Kalinin, "Static malware detection based on naive Bayes classifier and feature extraction," 2016 International Siberian Conference on Control and Communications (SIBCON). DOI: 10.1109/SIBCON.2016.7491688

[21] Z. Liu et al., "Malware detection based on deep learning," 2017 12th International Conference on Intelligent Systems Design and Applications (ISDA). DOI: 10.1109/ISDA.2017.8618779

[22] M. Santos and B. Santos, "Dynamic malware analysis: A survey," IEEE Communications Surveys & Tutorials, 2018. DOI: 10.1109/COMST.2017.2768262

[23] Pascanu et al., "On the difficulty of training recurrent neural networks," International Conference on Machine Learning, 2013. DOI: 10.1109/ICML.2013.170

[24] Y. Tian and T. Gong, "Malware clustering based on multi-behavior information," IEEE International Conference on Information Reuse and Integration, 2017. DOI: 10.1109/IRI.2017.61

[25] M. Almukhtar et al., "Hybrid approach for malware detection using static and dynamic analysis," 2021 International Conference on Electrical Engineering (ICEE). DOI: 10.1109/ICEE52809.2021.9465597

[26] R. Dhamija et al., "The battle against phishing attacks," Communications of the ACM, 2006. DOI: 10.1145/1188913.1188914

[27] M. Whittaker et al., "The prevalence of phishing on the web," Proceedings of the USENIX Security Symposium, 2010.

[28] A. Agarwal and V. Joshi, "Phishing detection methods: A survey," International Journal of Computer Applications, 2014. DOI: 10.5120/15615-4414

[29] M. Sahingoz et al., "Machine learning based phishing detection from URLs," Computers & Security, 2018. DOI: 10.1016/j.cose.2017.09.002

[30] T. Le and D. O'Hare, "A deep learning approach to website phishing detection," International Conference on Digital Forensics and Cyber Crime, 2017. DOI: 10.1109/ICDF2C.2017.00020

[31] N. Papernot et al., "Practical black-box attacks against machine learning," Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security. DOI: 10.1145/3052973.3052988

[32] T. Jaggi et al., "Challenges in Machine Learning-Based Security Systems," 2023 IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security). DOI: 10.1109/CyberSecurity53879.2023.10023807

[33] C. Szegedy et al., "Intriguing properties of neural networks," International Conference on Learning Representations, 2014. DOI: 10.1109/ICLR.2014.50617

[34] J. Goodfellow et al., "Explaining and harnessing adversarial examples," International Conference on Learning Representations, 2015. DOI: 10.1109/ICLR.2015.100

[35] Wang et al., "Few-shot learning: A survey," IEEE Transactions on Neural Networks and Learning Systems, 2022. DOI: 10.1109/TNNLS.2022.3180740

[36] M. T. Ribeiro et al., "Why should I trust you?" Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. DOI: 10.1145/2939672.2939778

[37] Q. Yang et al., "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology, 2019. DOI: 10.1145/3292042